



# Old Mutual - CSSA Computer Olympiad DAY 1



## Overview

<b>Problem</b>	<b>Count</b>	<b>Codes</b>	<b>Roads</b>
Program name	count.exe	codes.exe	roads.exe
Source name	count.pas	codes.pas	roads.pas
	count.java	codes.java	roads.java
	count.cpp	codes.cpp	roads.cpp
Input file	count.in	codes.in	roads.in
Ten (10) Output files	countn.out	codes.out	roads.out
Time limit per test	5 seconds	5 seconds	5 seconds
Number of tests	10	10	10
Points per test	10	10	10
<b>Total points</b>	<b>100</b>	<b>100</b>	<b>100</b>

The maximum total score for Day1 is 300 points.



# Old Mutual - CSSA Computer Olympiad Day 1



## Count (100) by Bruce Merry

### Description

In deepest Africa lives a tribe known as the Guji. The tribe has been expanding continuously for years and the chief is having difficulty keeping the tribal records up to date. He has employed an accounting student from UCT to help him with the records. However, they have run into problems because the Guji tribe does not use the decimal counting system.

In the decimal system, each digit is between 0 and 9, and represents a multiple of a power of 10. For example, 2301 represents  $2 \times 10^3 + 3 \times 10^2 + 0 \times 10^1 + 1 \times 10^0$ . In the Guji counting system, the  $n$ th number from the right (counting from 1) is between 0 and  $n$  and represents a multiple of  $n!$ , where  $n! = 1.2.3 \dots n$ . So for example 2301 would represent the number  $2 \times (1.2.3.4) + 3 \times (1.2.3) + 0 \times (1.2) + 1 \times (1)$ , and the number 2301 in decimal would be represented as 310311 in Guji notation.

### Task

To help the chief and the accountant, you must do a number of conversions between decimal and Guji numbering.

### Input (count1.in - count10.in)

You are provided with 10 input files, named count1.in, count2.in, ..., count10.in. Each file has the following format:

Line 1: Either the character 'D' or the character 'G'

Line 2: A number in either decimal or Guji.

If the first line is 'D' the number will be decimal and if the first line is 'G' it will be Guji.

### Sample input:

```
D
2031
```

### Output (count1.out - count10.out)

For each input file you are expected to produce an output file with the same name but the extension .out (so the output file for count3.in is count3.out). Note that only your output files are marked, not the program that you write (you may even choose to do the conversion by hand if you wish).

The output file consists of a single line, containing only the conversion of the number in the input file to the other counting system (i.e. if the input is in decimal, the output must be the same number in Guji and vice versa).

### Sample output:

```
244211
```

### Constraints

The number will be a positive integer less than or equal to 987654321 in the Guji numbering system (3628799 in decimal).

### Time Limit:

Maximum time per test case 5 seconds

### Scoring:

10 points for each correct output file.

0 points for each incorrect output file.



# Old Mutual - CSSA Computer Olympiad Day 1



## Codes (100) By Bruce Merry

### Description

The Guji tribe of deepest Africa have recently discovered some ancient writings in the walls of some caves. Although the Guji language has remained the same over the centuries, the way it is written has changed completely and the Guji are having difficulty reading it. They want you to write a program that can help them.

The ancient writing consists of groups of symbols, separated by dots. For the purposes of the program, the different symbols can be represented by the digits 0 to 9. A typical word might look like 2732.170.23.99. Each valid group has been replaced in modern Guji writing by a single letter from the English alphabet (you are given the mapping between the modern letters and the old groupings).

### Task

Unfortunately the passage of time has work away the dots and made the writing almost impossible to decipher! For example, the word above now appears as 27321702399 and it is impossible to tell where one grouping ends and the next begins. Your program is required to output every sequence of English letters that could correspond to the ancient word (you do not need to worry whether the sequence of letters forms a valid Guji word).

### Input (codes.in)

The first line contains the ancient form of the Guji word, as a sequence of digits (no dots) without any spaces separating them. The 2nd line contains N, number of letters from the English alphabet that the Guji use in their modern writing (they do not need the full alphabet). The remaining N lines each contain an English letter, then a space, then the symbol grouping that corresponds to that letter. All letters are upper case and no letter is repeated. However more than one letter may correspond to the same symbol group.

### Sample input:

```
27321702399
7
I 399
R 17023
T 99
U 21
J 702
G 273
A 2732
```

### Output(codes.out)

The output file should consist of a series of lines, each containing a possible English character string that might correspond to the ancient writing. The lines do not have to be in any particular order but no string should be duplicated. The letters should be in upper case.

### Sample output:

```
GUJI
ART
```

### Constraints:

Let L be the length of the ancient word, N be the number of valid groupings,  $G_i$  be the length of the  $i$ th valid grouping and S be the number of solutions. Then

```
1 <= L <= 200
1 <= N <= 26
1 <=  $G_i$  <= L
1 <= S <= 5000
```

### Time Limit:

Maximum time per test case 5 seconds

### Scoring:

One point is awarded for each distinct correct string. One point is removed for each distinct incorrect string. Two points are removed for every duplicate in the file. The score is then scaled so that a perfect solution gets 100%. Negative scores are converted to 0.

If any line of the output file contains letters or other characters not listed in the input file then an immediate score of 0 is given.



# Old Mutual - CSSA Computer Olympiad Day 1



## Roads (100)

### Description

The town planner of a small town that would prefer to stay nameless is planning a new tourist route through the town. She wants to introduce the tourists to all the beauties of the town, thus the route must travel over every road. She would also like to promote the businesses on both sides of each road. Thus the route must travel across each road in both directions (you may assume that there are no one-way roads). To avoid making the tour tedious, every road must be traversed exactly once in each direction. Furthermore, the tour must start and stop at the bus depot. Every road in the town is reachable from the bus depot. There are no roads that start and end at the same intersection and between any pair of intersections there can be at most one road.

### Task

The town planner has provided you with a road plan of the town. This plan contains information on all the intersections and roads in the town. The intersections are labeled by positive integers for simplicity. A list of roads is given. The description of a road is simply the numbers of the two intersections connected by the road. The bus depot is at intersection 1. (i.e. the tour must start and stop at intersection 1). Your task is to construct a suitable route for the town planner, given the layout of the roads in the town. There will always be a route that satisfies the requirements. There may be more than one appropriate route, but you are only required to find one.

### Input

The first line of the file 'ROADS.IN' contains the number of intersections  $N$  followed by the number of roads  $R$ . The next  $R$  lines contain the numbers of two intersections joined by a road on each line.

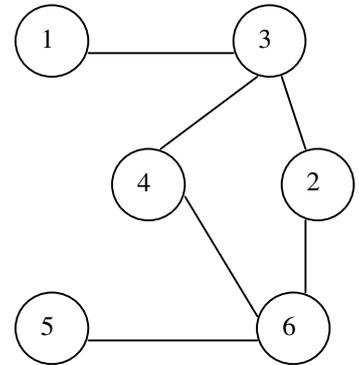
### Output

Your output file, 'ROADS.OUT', must contain in order the intersections visited by the tour, one intersection per line. Thus the first and last lines of the file should both contain '1'.

### Example

ROADS.IN

```
6 6
1 3
6 5
3 4
2 6
6 4
2 3
```



*explanation*

ROADS.OUT (possible solution)

```
1
3
2
6
5
6
2
3
4
6
4
3
1
```

### Constraints

$1 < N \leq 5000$   
 $1 \leq R \leq 32000$

### Time

Maximum time per test case 5 seconds

### Scoring

There will be 10 test cases, each of which will be weighted equally. Each test case can only score either 100% or 0%, depending on whether the solution is valid.